

Hybrid Quantum-Classical Autoencoders for End-to-End Radio Communication

Zsolt Tabi*, Bence Bakó*[†], Dániel T. R. Nagy*[‡], Péter Vaderna[‡], Zsófia Kallus[‡],
Péter Hága[‡], Zoltán Zimborás*[†]

*Eötvös Loránd University, Budapest, Hungary

Email: zsolttabi@ik.elte.hu

[†]Wigner Research Centre for Physics, Budapest, Hungary

Email: bako.bence@wigner.hu, zimboras.zoltan@wigner.hu

[‡]Ericsson Research, Budapest, Hungary

Email: {daniel.a.nagy, peter.vaderna, zsofia.kallus, peter.haga}@ericsson.com

arXiv:2301.02609v1 [quant-ph] 6 Jan 2023

Abstract—Quantum neural networks are emerging as potential candidates to leverage noisy quantum processing units for applications. Here we introduce hybrid quantum-classical autoencoders for end-to-end radio communication. In the physical layer of classical wireless systems, we study the performance of simulated architectures for standard encoded radio signals over a noisy channel. We implement a hybrid model, where a quantum decoder in the receiver works with a classical encoder in the transmitter part. Besides learning a latent space representation of the input symbols with good robustness against signal degradation, a generalized data re-uploading scheme for the qubit-based circuits allows to meet inference-time constraints of the application.

Index Terms—variational quantum algorithms, quantum machine learning, quantum autoencoder, radio communication

I. INTRODUCTION

One of the most popular Quantum Machine Learning (QML) methods are Quantum Neural Networks (QNNs) [1], [2]. These are special variational quantum circuits, designed as the quantum analogues of classical neural networks. QNNs can be optimized with gradient-based or gradient-free optimization algorithms forming hybrid quantum-classical training loops [3], [4]. Various QNN architectures have been proposed such as quantum convolutional neural networks [5], generative models [6], long short-term memories [7] and autoencoders [8]–[10]. Beside the high activity in algorithmic research within QML, their novel benchmarking and requirement setting applications are also motivating a wide variety of works [11].

Although QML is still in a phase of basic research with many open questions, its early implementations in wireless communication systems spark both scientific curiosity and commercial interest [12]. However, high-performance, near real-time applications might impose a new set of requirements on these solutions.

Wireless communication has undergone tremendous evolution during the last decades. The increasing adoption of AI and ML methods is opening up new development possibilities in various parts of the radio stack. In the design of the sixth generation (6G) wireless networks, AI and ML technologies are considered to be tightly integrated into the system and smart algorithms can be applied to all aspects of network

operations and procedures [13]. Considering the improvement of quantum computers it is envisioned that quantum algorithms and especially QML will play a significant role in future networks [12].

The structure of this paper is as follows. In Sec. II, we give a high-level overview of wireless communication systems together with an autoencoder solution used in the radio physical layer. Our novel hybrid classical-quantum autoencoder prototype is presented in Sec. III. We discuss our result in Sec. IV. Finally, we conclude with an outlook in Sec. V.

II. AUTOENCODER ARCHITECTURE IN END-TO-END COMMUNICATION

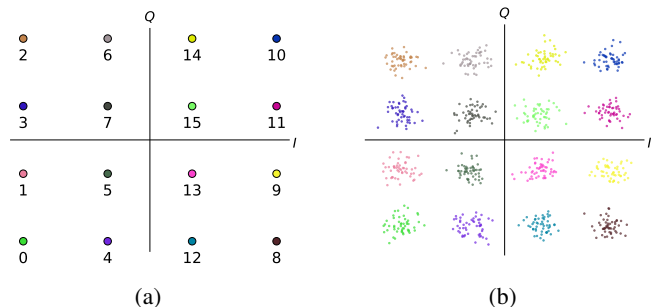


Fig. 1: **16-QAM constellation diagrams.** (a) Transmitted symbols; (b) Received noisy signal.

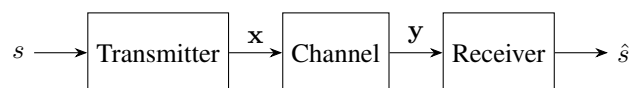


Fig. 2: **High-level representation of a communication system.** A message is transmitted through a noisy communication channel to be recovered by the receiver.

The components of communication networks are organized in a layered architecture where each layer is responsible for different communication aspects [14]. The physical layer is the

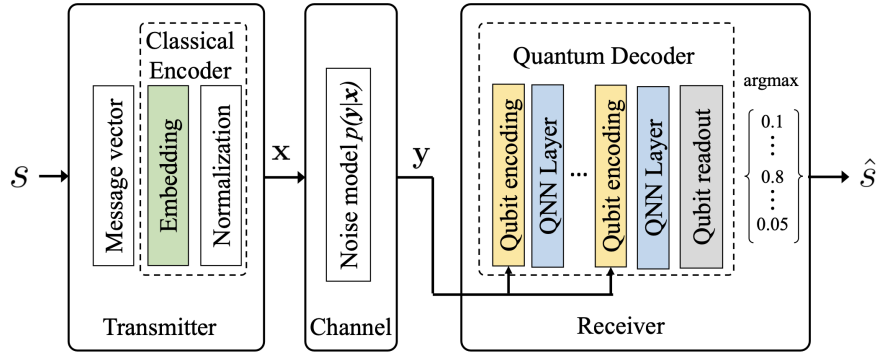


Fig. 3: **The proposed hybrid quantum-classical autoencoder embedded into the end-to-end communication architecture.** The transmitter maps each message s to a symbol, then sends it through the channel. The channel is represented by an additive noise acting on the signal x . The receiver, realized by a quantum decoder, consists of a multi-layer QNN adapting data re-uploading. It decodes the noisy signal and gives an estimate of the original message.

lowest layer. It provides the means of transmitting a stream of raw bits over a data channel connecting the network elements. The physical layer on the transmitter side converts data in the form of bits to electromagnetic waves to be transmitted wirelessly, while the receiver converts the electromagnetic waves received by an antenna to binary data. The main challenge in wireless data transmission is to overcome the channel impairments so that the messages can be recovered with small error rate.

The role of modulation is to convert digital data into radio waves. It can be achieved in different ways, the information can be encoded by varying (shifting) either amplitude, frequency or phase of the electromagnetic wave. The more states the modulation has, the more bits are transferred in one symbol, resulting in higher data rate. However, with higher order modulation the signal is more sensitive to channel errors, so the applied modulation usually depends on the channel quality. Fig. 1 shows a constellation diagram of the symbol representation of 16-QAM modulation, where 4-bit strings can be represented as complex numbers in a scheme resistant to general noise patterns while achieving high data rate with minimal channel uses.

Based on [15], we model a simple communications system, shown in Fig. 2, consisting of a *transmitter* that modulates message $s \in \mathcal{M} = \{0, \dots, M-1\}$ into a signal x and sends it over a noisy *channel* to the *receiver* that tries to decode the received signal y resulting in the received message \hat{s} . The transmitted signal x suffers degradation due to the noise present on the channel. In case of transmission over a complex channel with n discrete channel uses, the transmitter can be represented as the transformation $f : \mathcal{M} \mapsto \mathbb{R}^{2n}$, mapping the message s to $x \in \mathbb{R}^{2n}$ signal with certain constraints imposed by the transmitting hardware (e.g., energy constraint or average power constraint). The channel can be modeled as a conditional probability density function $p(y|x)$ that produces the output signal $y \in \mathbb{R}^{2n}$ given the input signal, usually via some noise model (e.g., additive white Gaussian

noise (AWGN)). The receiver is represented as the mapping $g : \mathbb{R}^{2n} \mapsto \mathcal{M}$ that recovers some estimate \hat{s} of the original message from the received signal. In this work, we focus on the case of single channel use ($n = 1$), however, this model can be easily adapted to cases of $n > 1$. Also, the number of transmit-receive pairs can be increased to get a Multiple-Input and Multiple-Output (MIMO) system [15], [16].

Both of f and g transformations can be created in various ways. In case of simple noise models applied in the channel the transformations can be designed as explicit mathematical formulas. In the case of complex noise scenarios that are difficult to describe with mathematical models, a possible way is to train deep neural networks, especially autoencoders, to solve the encoding and decoding tasks [15].

An autoencoder is a special type of deep neural network, with the aim to compress or denoise data [17]–[19]. Autoencoders consist of an encoder function $f : \mathbb{R}^D \mapsto \mathbb{R}^L$, and a decoder function $g : \mathbb{R}^L \mapsto \mathbb{R}^D$. The encoder transforms its input χ into a *latent space* representation $f(\chi) \in \mathbb{R}^L$, whereas the decoder tries to reconstruct it: $\hat{\chi} = g(f(\chi))$. Usually we have $L < D$, i.e., the encoder produces a compact representation of the data. f and g are typically deep neural networks trained jointly to minimize a loss function of the form $L(\chi, g(f(\chi)))$.

In telecommunication, opposite to the general compressing and denoising interpretation, autoencoders can be effectively used in the presented communication system to learn how to represent input messages as signals [15]. This model differs from the “typical” autoencoder concept in the sense that it does not try to remove noise from the input, instead it learns how to represent the input in a way that is robust against a given noisy channel acting in the latent space of the autoencoder. As a result of the training process, the latent space (or hidden layer) of the autoencoder contains the learned constellation of symbols (or codebook). The learned constellation is optimized for best mapping of the input messages to signals that can be accurately decoded with the largest success probability for the

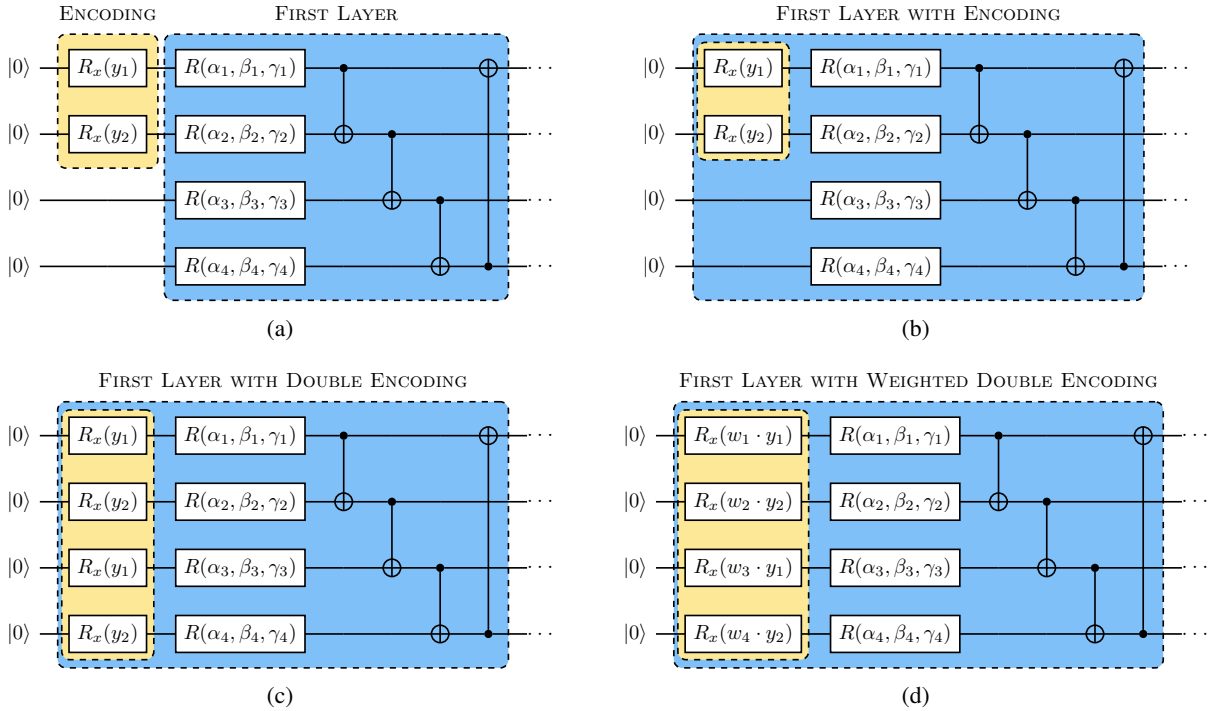


Fig. 4: **Quantum decoder implementations with encoding schemes.** Ansatz circuits with (a) simple data encoding; (b) simple data re-uploading; (c) double data re-uploading; (d) weighted double data re-uploading.

specific channel model. Whereas the encoder learns how to produce optimal symbols, the receiver learns how to decode these symbols after they have been corrupted by the channel, i.e., how to recover \mathbf{x} after sampling from $p(\mathbf{y}|\mathbf{x})$.

III. HYBRID QUANTUM AUTOENCODER FOR RADIO PHYSICAL LAYER

A. Hybrid quantum autoencoder overview

Quantum autoencoder architectures have previously been proposed to compress as well as denoise quantum data [8], [10]. Hybrid quantum-classical autoencoders enable many variations for quantum or classical encoding/decoding or the use of classical data. In this work, a hybrid quantum-classical autoencoder is applied for processing classical information. Building on the physical layer autoencoder presented in Sec. II, we propose a hybrid quantum-classical autoencoder with classical encoder on the transmitter side and a quantum decoder on the receiver side – trained in an end-to-end solution. The encoder projects the original message to a lower dimensional representation, robust to the channel degradation effect. Once the signal is passed to the quantum decoder, the compressed information is mapped to a higher dimensional Hilbert space of the qubits by a QNN that has been previously shown to be efficient for classification tasks [20], [21].

In our model, the classical encoder consists of an embedding followed by a normalization. A simple linear embedding is used to produce the constellation, satisfying the average power constraint by normalization. The decoder is realized by a general strongly connected quantum neural network which

we refer to as a *quantum decoder*. By simulating increasing levels of noise in the channel, we can present a performance evaluation of the various neural network architectures.

B. Quantum decoder architectures

A general QNN architecture has three main components as shown in Fig. 3: *qubit encoding* for embedding the input data, the parameterized *QNN layers*, and the *qubit readout* given as a probability distribution over the possible constellation symbols obtained from suitable measurements with high enough number of shots. To encode the output of the channel, we choose *angle embedding* with parameterized R_x rotations [22]. With this embedding, there are multiple ways to encode two-dimensional feature vectors into four qubits. As for the variational ansatz, we use strongly entangling layers introduced in quantum classifiers as they are known to be expressive reaching ‘wide corners of the Hilbert space’ [23]. The measurements are performed in the computational basis and the obtained probability distribution over the 16 basis states is the output of the decoder.

The simplest single-layer realization of such a QNN structure is presented in Fig. 4a. To improve this ansatz, we can apply the data re-uploading trick recently introduced in [24]. This technique, as shown in Fig. 4b, repeats the input encoding block before each layer of the QNN circuit. The intuition behind the effectiveness of this method is that by re-introducing the input before each layer, one can mimic the computational structure of typical classical deep neural networks, where the copying of the classical information is readily available, which

would be, without this trick, prohibited by the no-cloning theorem in quantum machine learning. The expressivity of a model can be further increased by applying the encoding on different subsystems in parallel [25]. With this in mind, we further enhance the ansatz by encoding the first feature into both qubit no. 1 and no. 3 and the second input feature into both qubit no. 2 and no. 4. This double data re-uploading ansatz is presented in Fig. 4c. As a final improvement, we considered the role of the number of trainable parameters. As the expressive power of the ansatz is highly dependent on the number of trainable parameters, one should try to include as many parameters as possible. One way to increase the number of parameters while keeping the circuit as shallow as possible – to respect the limited hardware capabilities and the inference time constraints of the application – is to introduce trainable weights in the data re-uploading blocks, as shown in Fig. 4d. This modification keeps the depth constant.

C. Training and fine-tuning

For our hybrid autoencoder to achieve low estimation errors, the training of the end-to-end system requires to be further improved via hyper-parameter tuning.

First, the training of the hybrid model is done on batches uniformly sampled from the set of messages $\{0, \dots, 15\}$. These are sent as two dimensional encoded symbols through the AWGN channel with SNR of 15 dB and i.i.d. noise.

The accuracy of the model is measured by evaluating the Symbol Error Rate (SER), a key performance indicator commonly used in radio communication. The network weight updates are calculated with the sparse categorical cross-entropy of the distribution generated by the decoder and the ground truth symbols. This loss function is used to calculate gradients in a mini-batch gradient descent with batch size of 64 and Adam optimizer [26]. We simulate the hybrid autoencoder using PennyLane [27], a quantum machine learning framework with its TensorFlow [28] backend. Second, we evaluate the reached model accuracy at various hyper-parameter settings. The search is conducted by KerasTuner [29] after partitioning the space as the simulator compute times are prohibitive of a full grid search.

We start by first evaluating the *learning rate* parameter set $\eta \in \{0.1, 0.01, 0.001\}$ using the simple ansatz presented on Fig. 4a with $L = 8$ layers with 1000-shot measurements and 1000 training steps. Based on these results, the only viable value of $\eta = 0.1$ is set for the rest of this study.

We continue with evaluating modifications to the basic ansatz but keeping the number of layers $L = 8$ and 1000 training steps fixed, to minimize the overall computation time. The results are shown in Fig. 5. For the basic circuit, the SER fluctuates around its initial value without showing convergence to a desirable level. A significant accuracy improvement of roughly 40% is achieved by implementing single data re-uploading (Fig. 4b with ansatz of $1 \times \text{DR}$). Introducing the double data re-uploading layer ($2 \times \text{DR}$ with ansatz of Fig. 4c) leads to another 15% improvement. Finally, we can even further increase the performance by another 20% when using the

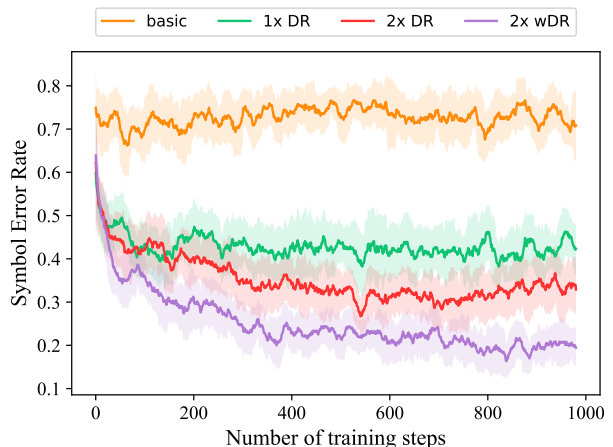


Fig. 5: **Learning curves of circuit architectures.** The number of data re-uploading (none, single, double) and the weighted data encoding have high impact on the convergence properties of the quantum autoencoder.

weighted double data re-uploading technique ($2 \times \text{wDR}$ with ansatz Fig. 4d). Based on these results, the weighted double data re-uploading ansatz is chosen for further experiments.

As a last step, we optimize the number of layers. The hybrid autoencoder using the best performing ansatz is trained with 8 to 24 layers. Increasing the number of layers clearly shows the improvement in SER as well as in convergence time as seen in Fig. 6.

IV. PERFORMANCE EVALUATION

A. Validation

Comparing our hybrid architecture to the classical method is crucial to validate the solution. Based on the learning curves presented, the shallowest network reaching accuracy similar to the classical solution contains $L = 16$ layers. Further increasing the number of layers leads to small improvements in accuracy but it is suboptimal in terms of circuit depth.

Although the hybrid quantum autoencoder models are trained at SNR of 15 dB we further validate the results at different values. The evaluation is shown in Fig. 7. We see that the trained networks generalize well on previously unseen SNR values, and reach performance similar to the classical baseline.

In Fig. 8, the constellation diagrams produced by autoencoders having different numbers of layers are shown. If the trained autoencoder has good performance, it is expected that the symbols are uniformly distributed in the diagram, similarly to Fig. 1. We see that increasing the number of layers leads to a more balanced distribution of symbols in the $Q - I$ space, which implies that the symbols can be well separated in case of noisy channels.

B. Time characteristics

In radio telecommunication, the latency of the data transmission is also an important performance metric. In some use

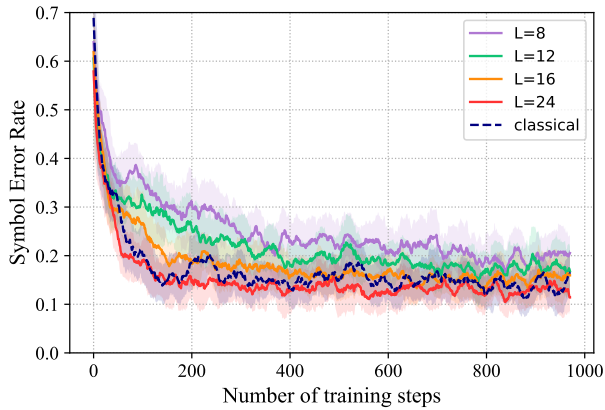


Fig. 6: **Learning curves of classical and hybrid autoencoders for a set of layer numbers.** We find that that the minimal number of layers necessary to achieve results comparable to the classical baseline is 16. Throughout these tests, we used ansatz according to Fig. 4d.

cases it is even critical that the end-to-end delay falls below a certain threshold. In 5G networks, it is possible to achieve ms level latency. Hence, in addition to the accuracy it is inevitable to investigate the time characteristics of the autoencoder model. After transpiling [30] the circuit ansatz to IBM QPU backend `ibmq_belem` and `ibmq_santiago` [31] and constructing the pulse-level scheduling, we can calculate the theoretical execution times on both QPUs. The transpiled circuits are deeper than the original ansatz, because we need SWAP gates due to limited qubit connectivity and the basis gate-set of the device can differ from the one used in Fig 4. In Table I, we present the circuit depth and the approximate per shot execution times of quantum decoders depending on the number of layers. The time values in the table suggest the following feasibility considerations for running QNN in a real-time system. The number of shots highly determines the reliability of the result of the inference. When the quantum decoder is executed with 1000 shots (a level already acceptable in current systems for this problem size), the inference time is the order of magnitude of $100ms$ which is higher than the accepted level in real-time radio systems. However, this can be reduced to the accepted level of below $10ms$ because the probability distribution is expected to be highly peaked for well-trained autoencoders.

V. CONCLUSION AND OUTLOOK

We presented a novel hybrid implementation of a quantum-classical autoencoder for end-to-end radio communication. The decoder was implemented as a variational quantum circuit. We showed that the use of advanced double re-uploading encoding schemes allows for the inference-time constraints of the application to be met without losing accuracy required from the autoencoder.

By implementing a combination of parallel encodings and weighted data re-uploading, we showed how these schemes

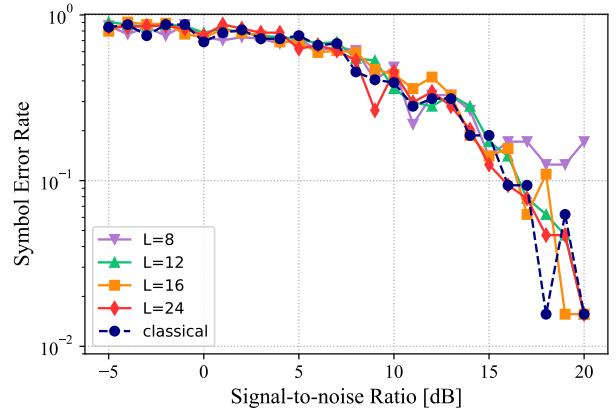


Fig. 7: **Validation of inference accuracy of the trained classical and hybrid autoencoders.** With increasing SNR values, hybrid models generalize to validation data on par with the classical.

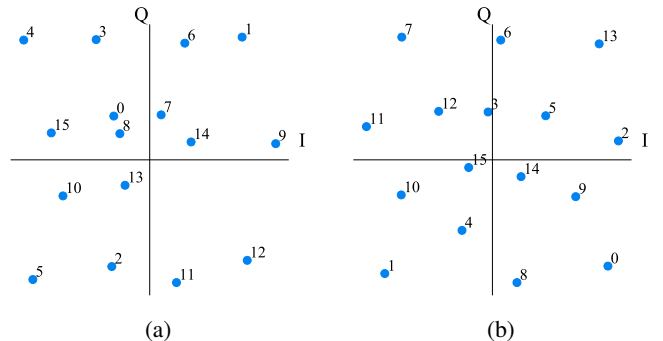


Fig. 8: **Constellations (latent space representations) learned by the hybrid autoencoder trained with SNR=15.** (a) $L = 8$ layers (b) $L = 24$ layers.

TABLE I: **Estimated execution times of the quantum decoder.** The circuit was run on the `ibmq_belem` and `ibmq_santiago` depending on the number of layers, calculated with Qiskit’s transpiler.

# layers	ibmq_belem		ibmq_santiago	
	depth	time [$\mu s/shot$]	depth	time [$\mu s/shot$]
8	125	54.3	145	30.4
12	187	78.4	221	43.6
16	260	111.8	297	56.9
20	311	124.2	373	70.12
24	379	149.8	449	83.4

can improve not just the QNN expressivity but also the performance of the whole autoencoder model. We expect these quantum-enhanced models to outperform classical ones in more complex channel noise scenarios, a direction for future study.

ACKNOWLEDGMENT

Zsolt Tabi and Zimborás Zoltán would like to thank the support of the Hungarian National Research, Development and Innovation Office (NKFIH) through the Quantum Information National Laboratory of Hungary and through the Grants No. FK 135220, K124351 and TKP2021-NVA-29.

REFERENCES

- [1] M. Schuld, I. Sinayskiy, and F. Petruccione, “The quest for a quantum neural network,” *Quantum Information Processing*, vol. 13, no. 11, pp. 2567–2586, Nov 2014. <https://doi.org/10.1007/s11128-014-0809-8>
- [2] N. Killoran, T. R. Bromley, J. M. Arrazola, M. Schuld, N. Quesada, and S. Lloyd, “Continuous-variable quantum neural networks,” *Phys. Rev. Research*, vol. 1, p. 033063, Oct 2019. <https://link.aps.org/doi/10.1103/PhysRevResearch.1.033063>
- [3] D. Wierichs, J. Izaac, C. Wang, and C. Y.-Y. Lin, “General parameter-shift rules for quantum gradients,” *Quantum*, vol. 6, p. 677, Mar. 2022. <https://doi.org/10.22331/q-2022-03-30-677>
- [4] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, M. Sohaib Alam, S. Ahmed *et al.*, “Pennylane: Automatic differentiation of hybrid quantum-classical computations,” *arXiv e-prints*, p. arXiv:1811.04968, Nov. 2018.
- [5] S. Wei, Y. Chen, Z. Zhou, and G. Long, “A quantum convolutional neural network on nisq devices,” *AAPPS Bulletin*, vol. 32, no. 1, p. 2, Jan 2022. <https://doi.org/10.1007/s43673-021-00030-3>
- [6] S. Lloyd and C. Weedbrook, “Quantum generative adversarial learning,” *Phys. Rev. Lett.*, vol. 121, p. 040502, Jul 2018. <https://link.aps.org/doi/10.1103/PhysRevLett.121.040502>
- [7] S. Y.-C. Chen, S. Yoo, and Y.-L. L. Fang, “Quantum long short-term memory,” in *APS March Meeting Abstracts*, ser. APS Meeting Abstracts, vol. 2021, Jan. 2021, p. V32.009.
- [8] C.-J. Huang, H. Ma, Q. Yin, J.-F. Tang, D. Dong, C. Chen *et al.*, “Realization of a quantum autoencoder for lossless compression of quantum data,” *Phys. Rev. A*, vol. 102, p. 032412, Sep 2020. <https://link.aps.org/doi/10.1103/PhysRevA.102.032412>
- [9] J. Romero, J. P. Olson, and A. Aspuru-Guzik, “Quantum autoencoders for efficient compression of quantum data,” *Quantum Science and Technology*, vol. 2, no. 4, p. 045001, aug 2017. <https://doi.org/10.1088/2058-9565/aa8072>
- [10] D. Bondarenko and P. Feldmann, “Quantum autoencoders to denoise quantum data,” *Phys. Rev. Lett.*, vol. 124, p. 130502, Mar 2020. <https://link.aps.org/doi/10.1103/PhysRevLett.124.130502>
- [11] S. Chakrabarti, R. Krishnakumar, G. Mazzola, N. Stamatopoulos, S. Woerner, and W. J. Zeng, “A Threshold for Quantum Advantage in Derivative Pricing,” *Quantum*, vol. 5, p. 463, Jun. 2021. <https://doi.org/10.22331/q-2021-06-01-463>
- [12] S. J. Nawaz, S. K. Sharma, S. Wyne, M. N. Patwary, and M. Asadzaman, “Quantum machine learning for 6G communication networks: State-of-the-art and vision for the future,” *IEEE Access*, vol. 7, pp. 46 317–46 350, 2019.
- [13] “6G – connecting a cyber-physical world - a research outlook toward 2030,” Ericsson, Tech. Rep., February 2022. <https://www.ericsson.com/4927de/assets/local/reports-papers/white-papers/6g-connecting-a-cyber-physical-world.pdf>
- [14] A. S. Tanenbaum and D. Wetherall, *Computer networks, 5th Edition*. Pearson, 2011. <https://www.worldcat.org/oclc/698581231>
- [15] T. O’Shea and J. Hoydis, “An introduction to deep learning for the physical layer,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, 2017.
- [16] D. García, J. O. Lacruz, D. Badini, D. De Donno, and J. Widmer, “Model-free machine learning of wireless siso/mimo communications,” *Computer Communications*, vol. 181, pp. 192–202, 2022. <https://www.sciencedirect.com/science/article/pii/S0140366421003704>
- [17] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006. <https://doi.org/10.1126/science.1127647>
- [18] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, <http://www.deeplearningbook.org>.
- [19] H. Neji, J. Noguera-Iso, J. Lacasta, M. Ben Halima, and A. M. Alimi, “Adversarial autoencoders for denoising digitized historical documents: The use case of incunabula,” in *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, vol. 6, 2019, pp. 31–34.
- [20] E. Farhi and H. Neven, “Classification with quantum neural networks on near term processors,” *arXiv: Quantum Physics*, 2018.
- [21] A. Mari, T. R. Bromley, J. Izaac, M. Schuld, and N. Killoran, “Transfer learning in hybrid classical-quantum neural networks,” *Quantum*, vol. 4, p. 340, Oct. 2020.
- [22] M. Schuld and F. Petruccione, *Machine Learning with Quantum Computers*, ser. Quantum Science and Technology. Springer International Publishing, 2021. <https://books.google.hu/books?id=N5IEAAAQBAJ>
- [23] M. Schuld, A. Bocharov, K. M. Svore, and N. Wiebe, “Circuit-centric quantum classifiers,” *Phys. Rev. A*, vol. 101, p. 032308, Mar 2020. <https://link.aps.org/doi/10.1103/PhysRevA.101.032308>
- [24] A. Pérez-Salinas, A. Cervera-Lierta, E. Gil-Fuster, and J. I. Latorre, “Data re-uploading for a universal quantum classifier,” *Quantum*, vol. 4, p. 226, 2020.
- [25] M. Schuld, R. Sweke, and J. J. Meyer, “Effect of data encoding on the expressive power of variational quantum-machine-learning models,” *Phys. Rev. A*, vol. 103, p. 032430, Mar 2021. <https://link.aps.org/doi/10.1103/PhysRevA.103.032430>
- [26] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [27] V. Bergholm, J. A. Izaac, M. Schuld, C. Gogolin, and N. Killoran, “Pennylane: Automatic differentiation of hybrid quantum-classical computations,” *ArXiv*, vol. abs/1811.04968, 2018.
- [28] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro *et al.*, “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015, software available from tensorflow.org. <https://www.tensorflow.org/>
- [29] T. O’Malley, E. Bursztein, J. Long, F. Chollet, H. Jin, L. Invernizzi *et al.*, “Kerastuner,” <https://github.com/keras-team/keras-tuner>, 2019.
- [30] M. S. ANIS, Abby-Mitchell, H. Abraham, AduOffei, R. Agarwal, G. Agliardi *et al.*, “Qiskit: An open-source framework for quantum computing,” 2021.
- [31] “IBM quantum,” 2021. <https://quantum-computing.ibm.com/>